06-30-00   A/PROV

Attorney's Docket No. 18360/195005
Express Mail Label No. EL149284362US

PATENT

COVER SHEET FOR FILING PROVISIONAL PATENT APPLICATION

Box Provisional Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

This is a request for filing a PROVISIONAL PATENT APPLICATION under 37 C.F.R. 1.53(c).

Docket No.	—
Type a plus sign (+) inside this box →	+

INVENTOR(s)/APPLICANT(s)

Name: Cindy Johnson
400 Northridge Road, Suite 1000
Atlanta, Georgia 30350

Name: Will Fastie
Address: 6525 Best Friend Road
Norcross, GA 30071

Name: A.J. Wilson
Address: 990 Hammond Drive
Atlanta, GA 30328

Name: Tim Zack
Address: 990 Hammond Drive
Atlanta, GA 30328

TITLE OF THE INVENTION *(280 characters maximum)*

METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR REAL-TIME SUPPLY CHAIN MANAGEMENT

CORRESPONDENCE ADDRESS

Gregory T. Gronholm
Registration No. 32,415
Customer Number 000826

Tel. Atlanta Office (404) 881-7000
Fax Atlanta Office (404) 881-7777

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification (Number of Pages 42)
□ Drawing(s) (Number of Sheets ___)
□ Claims (Number of Claims ___)
(A complete provisional application does not require claims 37 C.F.R. § 1.51(a)(2).)

☐ Small Entity Statement
☐ Other (specify)

METHOD OF PAYMENT *(check one)*

☐ Check or money order is enclosed to cover the filing fee.

☒ The Commissioner is hereby authorized to charge filing fees and credit Deposit Account No. 16-0605.

☒ Please charge Deposit Account No. 16-0605 for any fee deficiency.

PROVISIONAL FILING FEE AMOUNT(s)

Large Entity $150.00
Small Entity $ 75.00

Filing Fee Amount: $150.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

Gregory T. Gronholm
Registration No. 32,415
Date: June 29, 2000

ALSTON & BIRD LLP
Post Office Drawer 34009
Charlotte, NC 28234
Tel. Atlanta Office (404) 881-7000
Fax Atlanta Office (404) 881-7777

CERTIFICATE OF EXPRESS MAIL

"Express Mail" mailing label number EL149284362US
Date of Deposit: June 29, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Box: Provisional Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Gregory T. Gronholm

METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR REAL-TIME SUPPLY CHAIN MANAGEMENT

FIELD OF THE INVENTION

The present invention relates to a supply chain management system, and more particularly, to a supply chain management system that processes requests for goods in at least near real-time such that specific goods can be reserved in response to the requests.

BACKGROUND OF THE INVENTION

Promising has in the past been constrained by the batch processing nature of ERP systems. In batch processing promising, a promise can not be delivered to the customer at the time an order is submitted because the order is merely accepted but not processed against the most current known inventory, warehouse capability and transportation availability until the next batch processing. This leads to coarse initial promising (*e.g.* Order No. 32987 comprising two widgets will be shipped to customer between 48 to 72 hours in the future) with significant adjustment notices after the batch processing or delayed promising. The invention provides a system approaching real-time promising, particularly to address the Business-to-Consumer transaction space for Internet based orders.

DESCRIPTION ACCORDING TO ONE PREFERRED EMBODIMENT OF THE PRESENT INVENTION

The present invention implements an at least near real-time supply chain system by offering solutions relating to the offering of more immediate and more accurate promises in response to requests for goods. Some of these solutions relate to inventory supply and demand, warehouse capacity, promising of stock inventory and inventory yet to be received, shipping and delivery, and like logistics considerations. These solutions, as well as the need for real-time supply systems, are set forth in the following three documents, attached hereto:

(1) A 19 (nineteen) page document identifying the at least near real-time supply chain system of the present invention.

(2) A 3 (three) page outline listing of features of the at least near real-time supply chain management system of the present invention. The outline discusses the manner in which the present invention provides solutions to problems such as inventory control, and the generation of promises to fulfill inventory requests.

(3) A 16 (sixteen) page detailed summary describing order promising features of the at least near real-time supply chain management system of the present invention. The document begins with a Table of Contents, and includes multiple chapters on specific elements of the supply system.

According to one novel embodiment of the present invention, a system is disclosed for facilitating the at least near real-time fulfillment of order requests. The system includes a promising engine that receives an order request from an ordering entity, such as a consumer or business. In response to the order request, the promising engine queries one or more databases associated with a plurality of warehouses to determine if the request can be fulfilled by inventory existing in, or scheduled to be received by, one of the plurality of warehouses. According to one aspect of the invention, the promising engine can query the databases based upon order fulfillment rules, where the order fulfillment rules determine the order in which the promising engine queries the plurality of databases.

According to another novel aspect of the present invention, the fulfillment rules direct the promising engine to query a database associated with a warehouse located nearest a shipment destination associated with the order request. In this manner, the present invention initially attempts to fulfill the order from warehouses located near the order request such that shipment costs will be minimized. If the warehouse located nearest the shipment destination is unable to reserve the requested goods, the fulfillment rules then queries the database associated with the next-closest warehouse. This process will continue until a warehouse can reserve inventory to fulfill the order. By polling the closest geographic warehouse first, the present invention assumes that order fulfillment from the nearest possible geographic warehouse is the most efficient means for fulfilling an order.

The promising engine typically responds to an order request by reserving a requested good(s) from a particular warehouse. Furthermore, based upon shipment criteria set by the ordering entity and/or promising engine, the promising engine can promise the good(s) to the ordering entity on a particular day, or within a particular timeframe. As compared with prior art batch promising, the system of the present invention bases promises on at least near real-time inventory updates received by databases associated with warehouses. Thus, the promising engine can promise particular goods reserved by the engine in response to a request, rather than making a promise and thereafter attempting to preserve the promise.

According to one aspect of the present invention, however, the promising engine will refrain from reserving goods requested by an ordering entity where the request for delivery is past a promising horizon of the promising engine, which is the length of time into the future that at least near real-time promising is able to execute. The promising horizon of the promising engine is a set timeframe past which the promising engine will not reserve goods. For instance, where an ordering entity requests receipt of goods six months in the future, the present invention recognizes that it is inefficient for the promising engine to reserve the goods for that length of time, as warehouse capacity must be increased, and immediate requests may be sacrificed to fulfill requests that can be met by goods received by one or more warehouses at a much later date. Therefore, the promising engine will automatically promise the requested good to the ordering entity at the requested time, and will later reserve a good to fulfill the promise at a time within the promising horizon. For instance, where the promising horizon is 21 days, and an order is requested for delivery 30 days from the date of the order request, the promising engine will immediately promise the goods for delivery 30 days after the order request, but may only reserve the goods after another 9 days have passed.

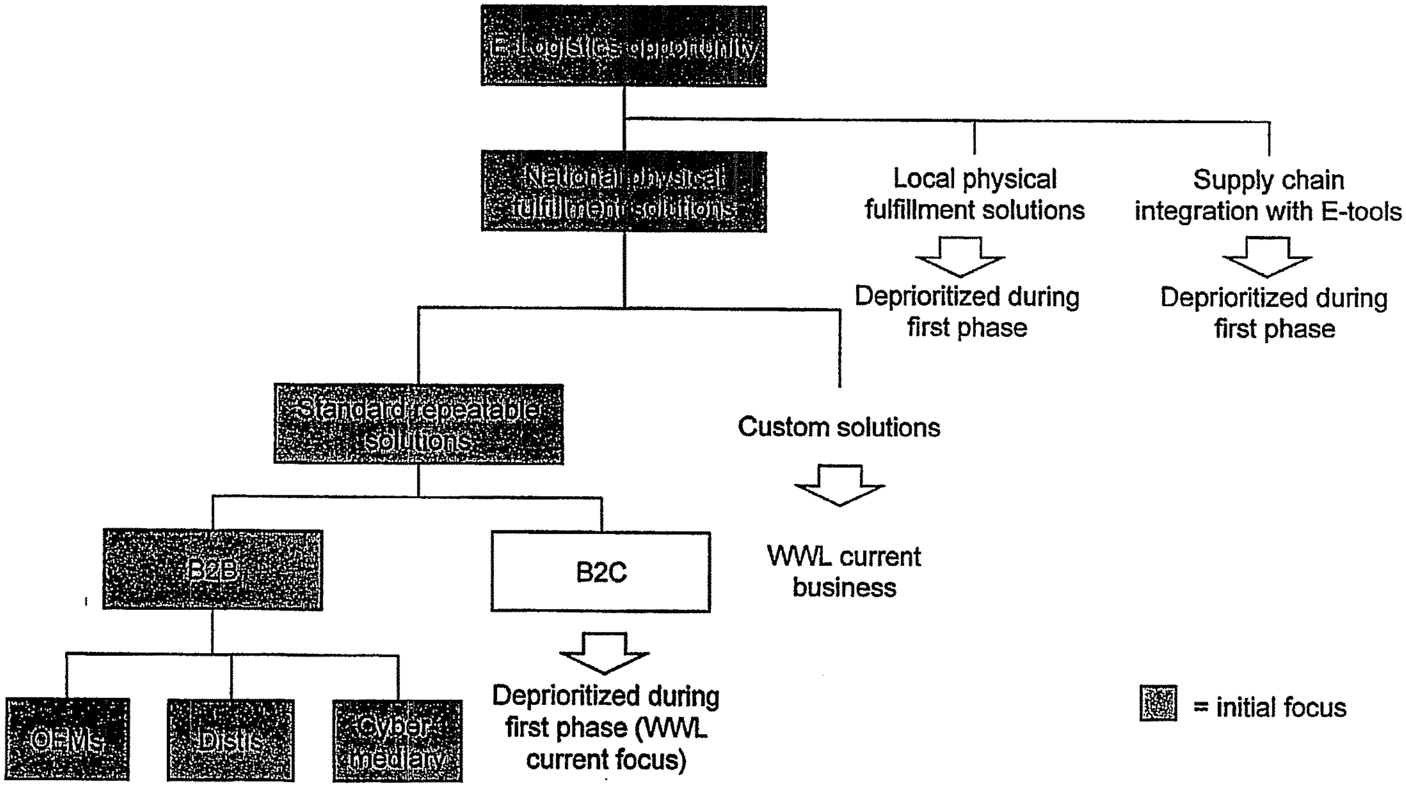
According to another aspect of the invention, the databases associated with the warehouses can consider inventory yet to be received. In this manner, the databases can determine not only current inventory, but inventory to be received by the warehouse, facilitating warehouse logistical planning. Inventory yet to be received can also be utilized by the databases or promising engine in determining the particular warehouse from which an order is to be fulfilled. For example, where one warehouse has a sufficient supply of inventory for an item, the system of the present invention may nonetheless use an alternative warehouse to supply an item where the alternative warehouse has a large incoming shipment and insufficient space to store the shipment, such that clearance of goods may be necessary. It will be appreciated by those of skill in the art that any variety of supply, demand, and transportation logistical considerations may be utilized by the promising engine to determine the location from which an order is to be fulfilled. According to yet another illustrative example, the promising engine may fulfill the order from an alternative warehouse (alternative used to describe a warehouse other than the warehouse located geographically nearest the shipment destination) where shipment cannot be made from the geographically nearest warehouse due to transportation problems.

According to one embodiment of the present invention, the system can be implemented such that diverse clients can offer goods via the system of the present invention without customizing the system. For instance, the at least near real-time fulfillment system can provide the order receipt, fulfillment, supply and transportation backbone to diverse businesses transacting on the Internet. Similarly, according to one aspect of the present invention, the system can include a plurality of geographically distant warehouses to fulfill order requests, where each warehouse maintains inventory and ships goods sold by a plurality of different businesses. Because customers interact with the business's Internet sites, they will be unaware that the system of the present invention can collectively store dissimilar goods in one warehouse to achieve economies of scale and to minimize costs associated with warehouse space, shipment, and warehouse management.

Agenda

- Background for creation of e-Ventures group
- E-Screening
- E-Logistics
- E-Commerce Incubator

Potential Opportunities



E-Logistics opportunity

National physical fulfillment solutions

Local physical fulfillment solutions

Deprioritized during first phase

Supply chain integration with E-tools

Deprioritized during first phase

Standard repeatable solutions

Custom solutions

WWL current business

B2B

B2C

Deprioritized during first phase (WWL current focus)

OEMs

Distis

Cyber-mediary

▨ = initial focus

Project Objectives

- Determine full potential UPS opportunity to provide logistics/fulfillment services for web-based commerce
  - business to business
- Identify scope of services and target markets for E-logistics fulfillment service
- Understand potential UPS service delivery cost position and determine whether UPS can make a sustainable profit and win
- Assess IT practicality in addition to cost position
- Determine appropriate resource investments, develop a business plan and gain organizational consensus to capture the opportunity

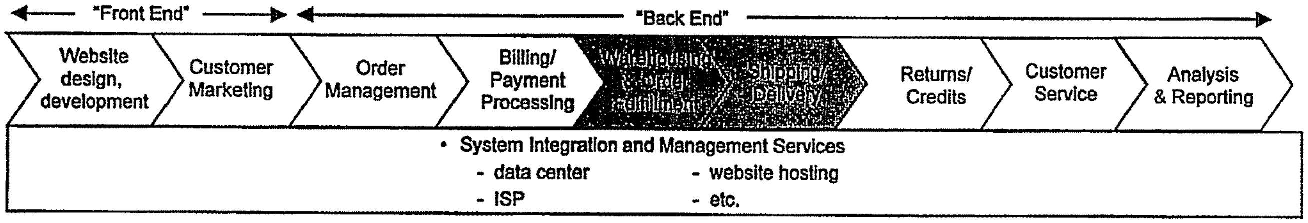
Service Concept

Potential Service Bundle

PRELIMINARY

Concept is for a turnkey service bundle enabling organizational segregation and rapid launch of e-commerce business units/startups.

Services:



- System Integration and Management Services
  - data center
  - ISP
  - website hosting
  - etc.

Website design, development	Customer Marketing	Order Management	Billing/ Payment Processing	Warehousing & Order Fulfillment	Shipping/ Delivery	Returns/ Credits	Customer Service	Analysis & Reporting
• Web site development and maintenance • Web hosting • Content development and management	• Direct Marketing • Tele-marketing • Database management	• Customer verification • Address verification • Fulfillment instructions • Real-time inventory status	• Credit authorization • Bill payment processing • Collections • Financing services	• Receiving • Stocking • Pick and pack • Inventory management and reporting • Inventory financing	• Shipping • Shipment confirmations • Track and trace	• Returned inventory tracking • Repacking • Stocking • Credit issuance	• Call centers • Order/ customer inquiry • Order modifications	• Tax and regulatory • Service level metrics

Infrastructure description:

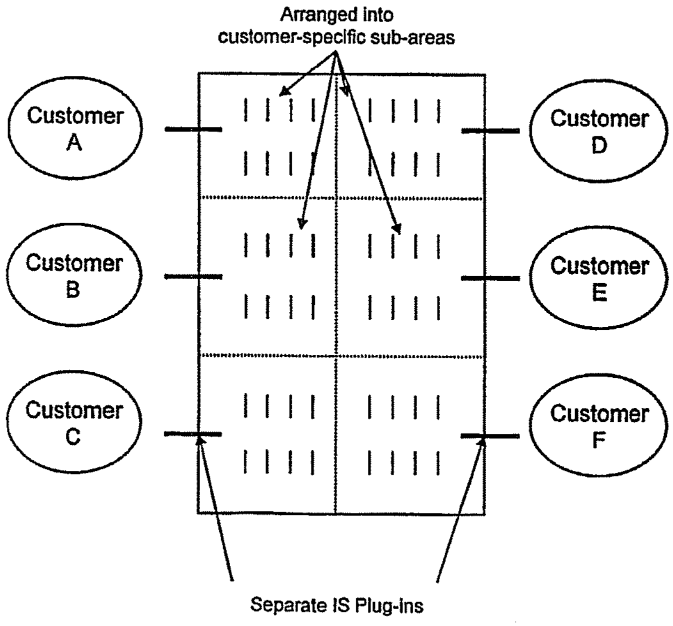
- Create standard warehouse/distribution solutions to service the U.S./North America at a low cost leveraging existing UPS WWL infrastructure and assets where possible
- Partner with 3rd party providers as necessary for service offerings such as website design, teleservices, etc.

Service Concept

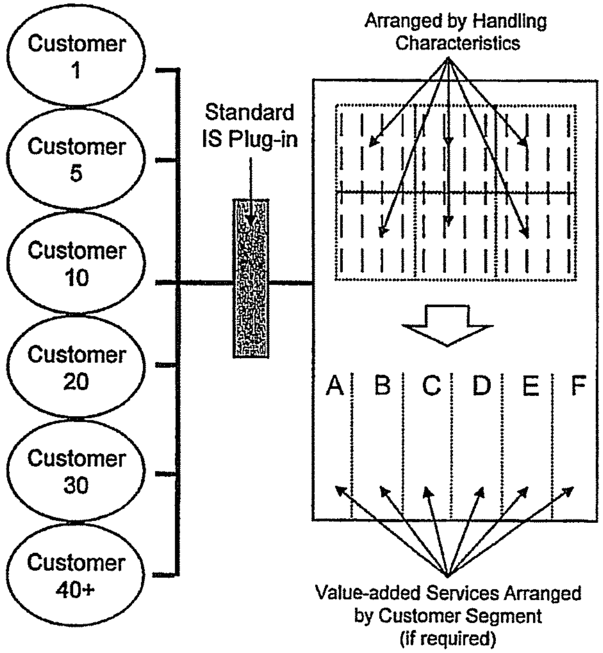
Warehouse Layout

ILLUSTRATIVE

Current WWL Model



E-Logistics Model


Customer 1
Customer 5
Customer 10
Customer 20
Customer 30
Customer 40+
Standard IS Plug-in
Arranged by Handling Characteristics
A
B
C
D
E
F
Value-added Services Arranged by Customer Segment (if required)

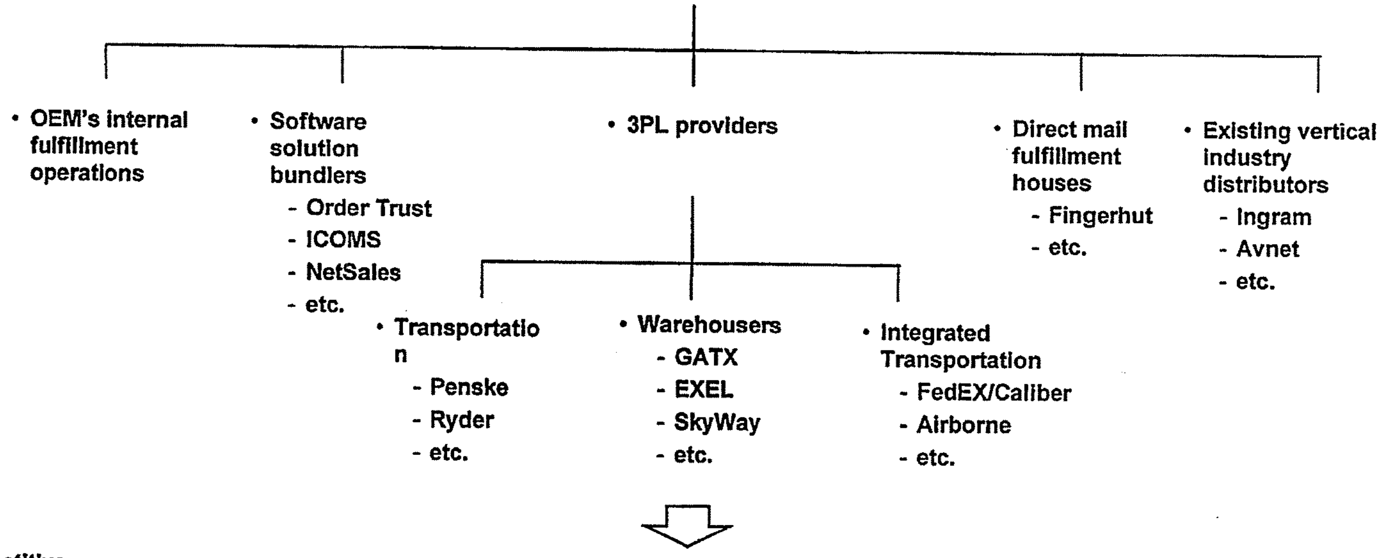
Potential Customers for Service

- Dot.com divisions of large corporations
- Dot.com startups
- Web front ends for sub-scale distributors
- Cybermediaries

E-Logistics

Competition

- E-Commerce “virtual business turnkey” offering will compete with at least 7 alternatives



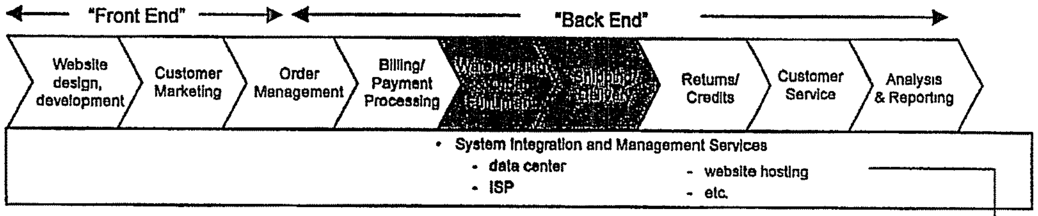
- OEM's internal fulfillment operations
- Software solution bundlers
  - Order Trust
  - ICOMS
  - NetSales
  - etc.
- 3PL providers
  - Transportatio n
    - Penske
    - Ryder
    - etc.
  - Warehousers
    - GATX
    - EXEL
    - SkyWay
    - etc.
  - Integrated Transportation
    - FedEX/Caliber
    - Airborne
    - etc.
- Direct mail fulfillment houses
  - Fingerhut
  - etc.
- Existing vertical industry distributors
  - Ingram
  - Avnet
  - etc.

Competitive Threat:							
-B2B	Weak	Weak	Moderate	Moderate	TBD	Moderate	Strong
-B2C	Weak	Weak	Weak	Moderate	TBD	Strong	Varies by Category

E-Logistics

Competitor Capability

○ Weak
◒ Moderate
● Strong

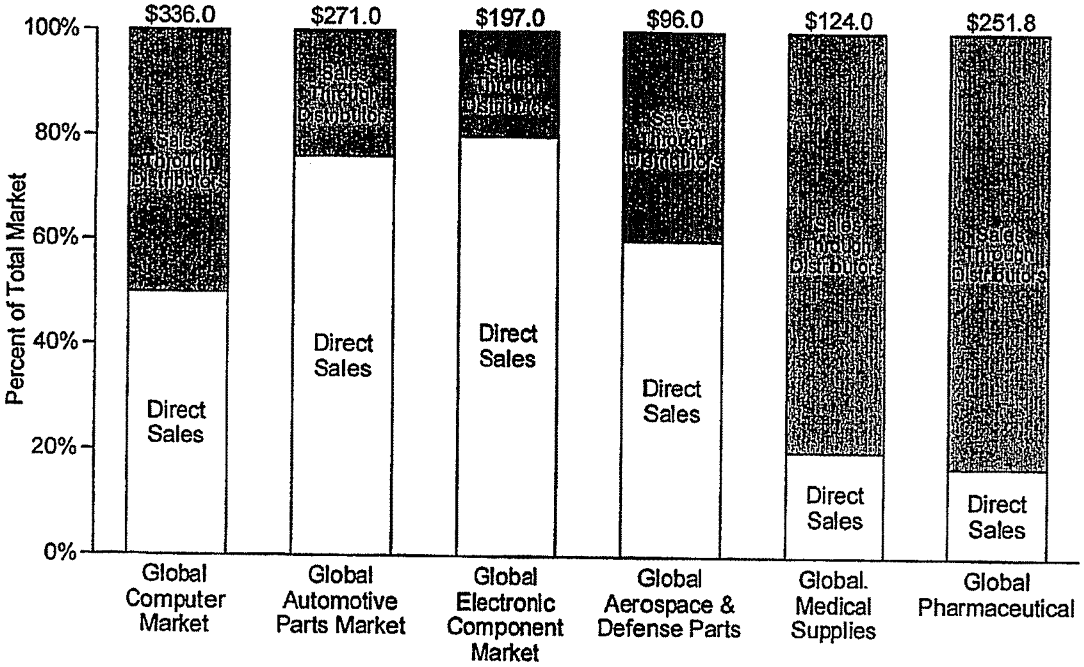


- System Integration and Management Services
  - data center
  - ISP
  - website hosting
  - etc.

	Website design, development	Customer Marketing	Order Management	Billing/ Payment Processing	Warehousing/ Fulfillment	Shipping/ Delivery	Returns/ Credits	Customer Service	Analysis & Reporting	System Integration and Management Services	**Overall**	Competitive Threat Given Focus **B2B**	 **B2C**
OEM's	○	◒	◔	◔	◔	◔	◔	◒	◒	○	○	Weak	Weak
Software Solution Providers	○	○	●	●	○	○	○	○	●	●	○	Weak	Weak
Transportation	○	○	◒	○	○	●	○	○	◒	○	◒	Moderate	Weak
Warehousers	○	○	◒	○	●	◒	○	○	◒	○	◒	Moderate	Moderate
Integrated Transportation												TBD	TBD
Fulfillment Houses	◒	●	●	●	●	◒	●	●	●	◒	●	Moderate	Strong
Distributors	○	◒	●	●	●	◒	●	●	●	◒	●	Strong	Vanes

Direct vs. Indirect Distribution

Distributors play a significant role in fulfillment of goods across vertical industries.



Source: Electronic News, Automotive News,U.S. Industry Trade & Outlook, Boelng, AGES, Aerospace Industries Association

Competitive Summary

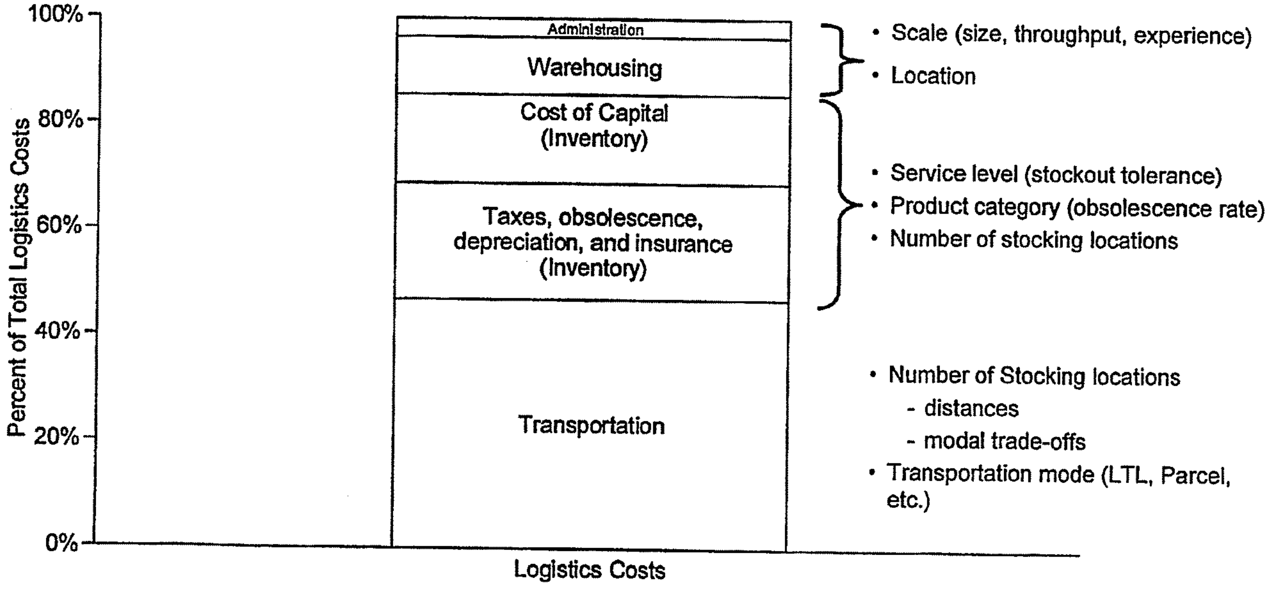
- Competition to beat is likely to be the distributors
  - motivated by threat of business loss
  - some already taking steps to develop commercial pilots

- Challenges to overcome include:
  - significant role in non-ecommerce sales (OEMs may fear retaliation if disintermediate)
  - existing scale warehouses and handling processes

- Advantages for UPS to prove and develop:
  - unbiased -- enthusiastic supporter of manufacturer internet initiatives
  - lower total cost solution due to warehouse placement and transportation savings
  - superior service levels (speed of delivery and accuracy) to enhance manufacturer brand
  - broaden scope of services to facilitate time to market on internet

Logistics Costs Components

UPS's total cost position will depend on combined network economics of transportation, inventory holding and warehousing versus alternative offerings from distributors, etc.



Drivers

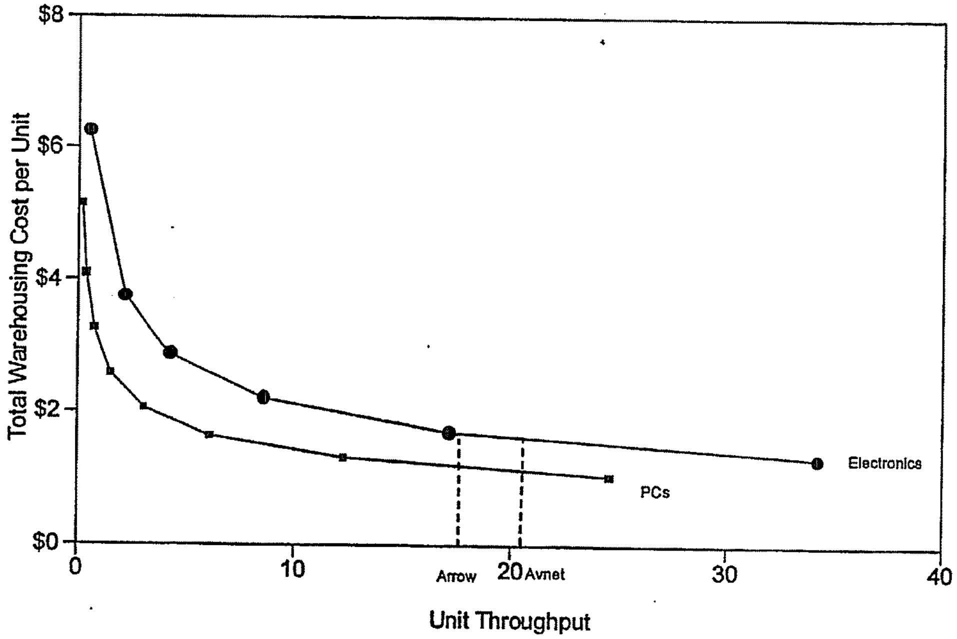
- Scale (size, throughput, experience)
- Location

- Service level (stockout tolerance)
- Product category (obsolescence rate)
- Number of stocking locations

- Number of Stocking locations
  - distances
  - modal trade-offs
- Transportation mode (LTL, Parcel, etc.)

Warehouse Scale Economics

Throughput economies begin flattening noticeably at 5M-10M units per year or about $10M-$20M of warehousing revenue.

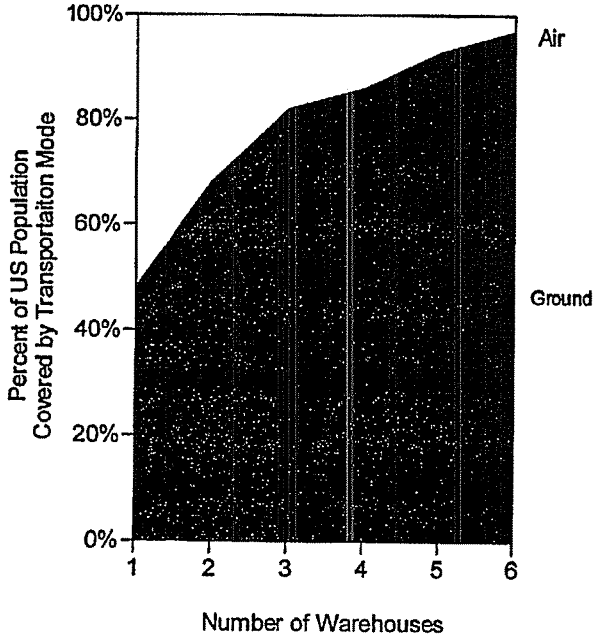


Note: Maintains WWL fixed/variable split for theoretical curve

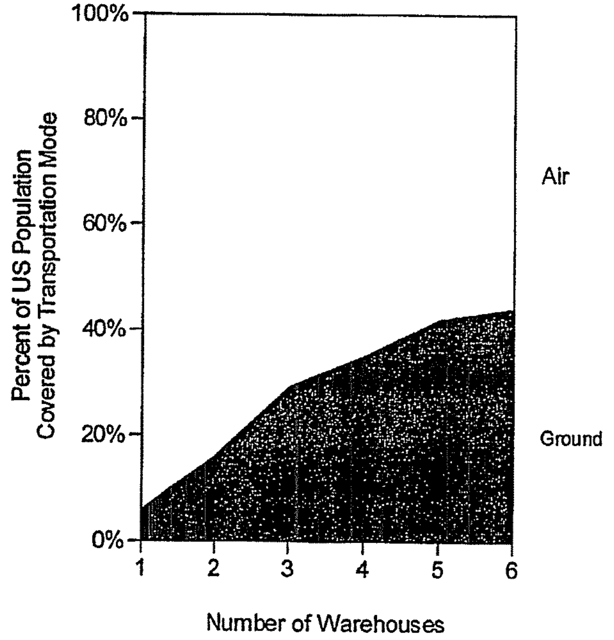
Transportation Mode and Coverage

PRELIMINARY

Two Day Service



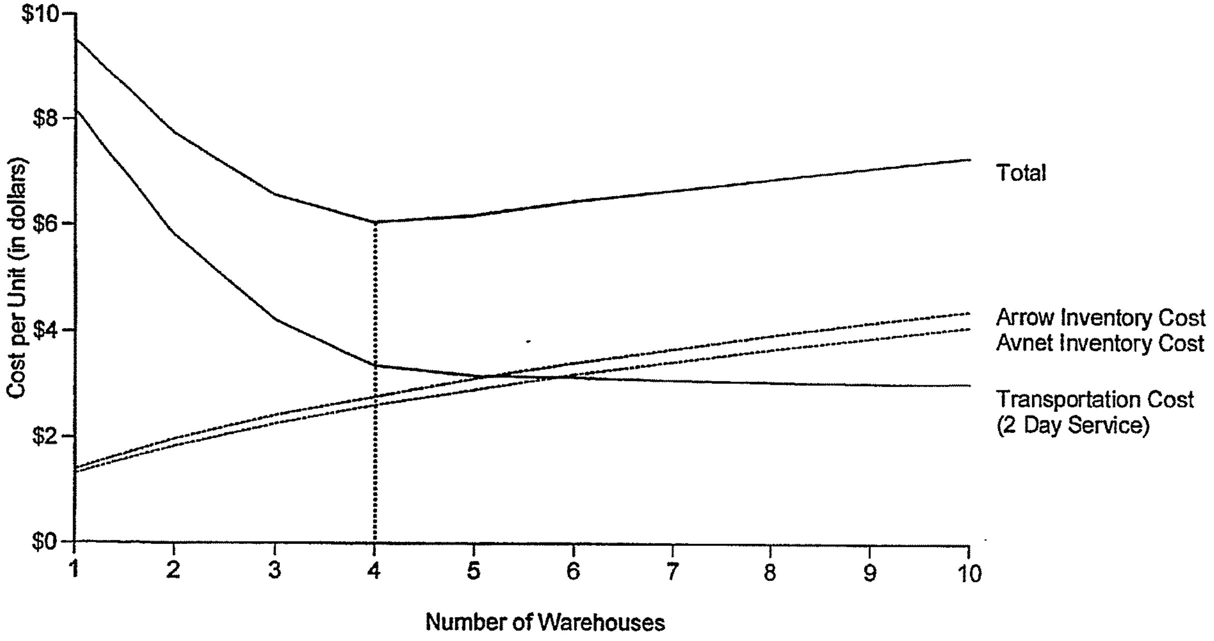
One Day Service



Note: Next day service based on 200 mile ground delivery radius (preliminary working figure - provided by UPS)
Source: Bain Analysis; PSI Analysis (Logic_Net);

Warehouse Solution: Electronic Components (1)

With 4-5 warehouses, UPS can cost effectively serve electronic component e-clients.



Note: Based on 12% WACC, 0.25% Insurance, 0.50% Tax, 5% Depreciation, and 0% Obsolescence;
Transportation estimate based on 2 units per package

Relative Cost Summary

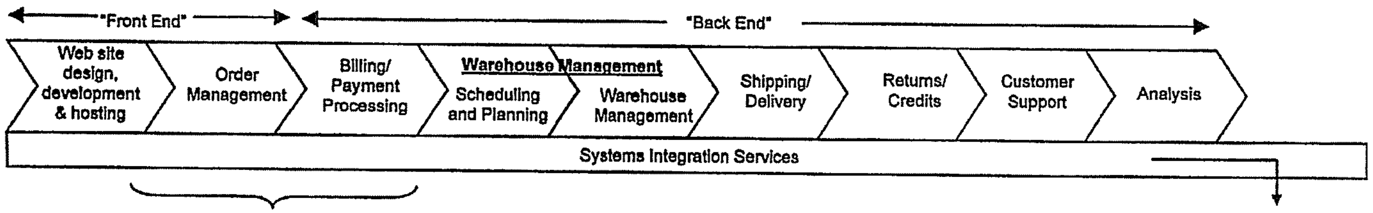
- **Total logistics costs will dominate potential customers' cost**

- **Initial analysis supports cost viability of concept**
  - **warehousing scale parity achievable at reasonable revenues**
  - **outbound transportation advantaged due to sites, especially for 1-day service** (and based on published rates)

- Remaining issues to resolve:
  - true UPS network costs vs. published rates
  - second-order warehouse cost effects (complexity costs, learning curve, value added processes)
  - inventory impact of relative total volumes

E-Logistics IT

Best-in-class Vendors (Gartner Perspective)

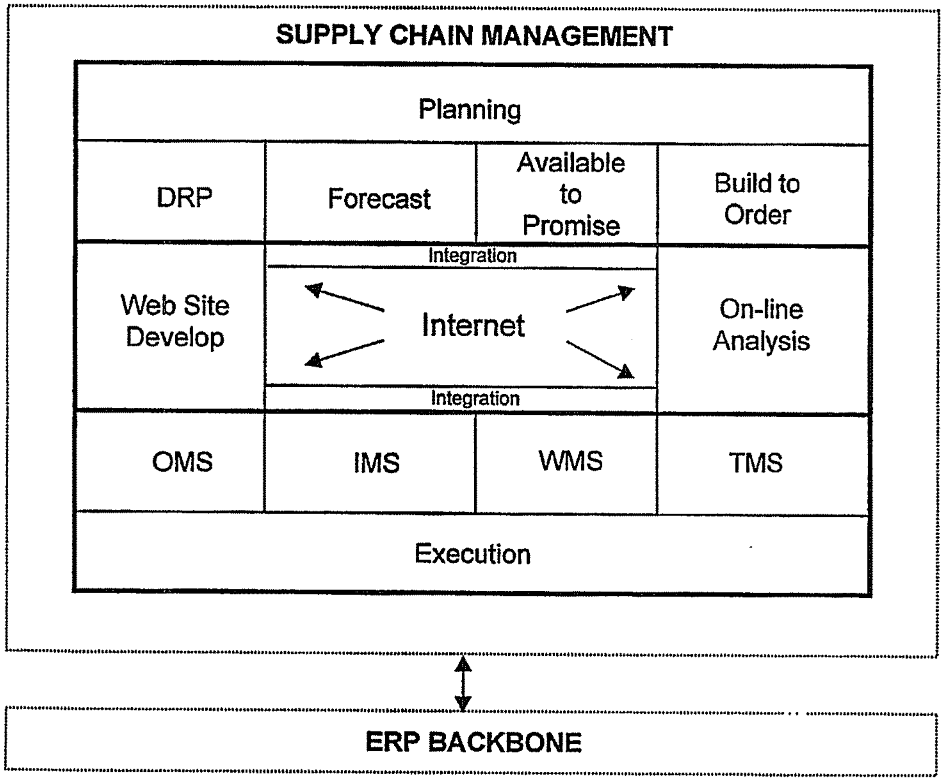


Web site design, development & hosting	Order Management / Billing/ Payment Processing	Scheduling and Planning	Warehouse Management	Shipping/ Delivery	Returns/ Credits	Customer Support	Analysis	Systems Integration Services
• BroadVision • iCat • IBM • Microsoft • Netscape (AOL) • Open Market	• IMI • CyberCash • CyberSource • ICat • IBM • Open Market • Verifone (Hewlett Packard)	• i2 • ILOG • Logility • Manugistics • Numetrix	• EXE Technologies • HK Systems • Manhattan Associates • McHugh Software • Yantra	• CAPS Logistics (Baan) • Descartes • I2 • Manugistics	• EXE Technologies • HK Systems • Manhattan Associates • McHugh Software • Yantra	• TBD	• TBD	• BEA Software • i2 • System Integrators - Andersen - CAP Gemini - CSC - KMPG • TRW • Yantra

Source: Gartner Group Interviews

E-Logistics IT

Software Architecture



SUPPLY CHAIN MANAGEMENT

Planning

DRP

Forecast

Available to Promise

Build to Order

Web Site Develop

Integration

Internet

Integration

On-line Analysis

OMS

IMS

WMS

TMS

Execution

ERP BACKBONE

E-Logistics IT

Strategic Options

Several strategic options exist to obtain an IT solution.

	Develop Internally	Knit Together Through System Integrator or Software Vendor	Acquire Components Through Acquisition
Strategy:	• Leverage WWL's IT experience and resources to build scaleable solution	• Contract with systems integrator or software vendor to develop IT solution	• Acquire IT solution through acquisition of a third party
Execution:	• Assign team to evaluate and develop IT solution	• Identify and interview capable IT vendors and partner to create IT solution	• Identify acquisition targets who posses necessary IT capability - OrderTrust - ICOMS
Issues:	• What are the internal IT capabilities and can they be leveraged to create a scaleable IT solution? • What additional resources will be needed and at what cost?	• What vendor(s) can provide the most comprehensive end-to-end, scaleable IT solution? • How much customization is needed and at what cost?	• Can the acquisition targets be acquired at a fair price? • Does their IT solution fit with E-Logistics' requirements?
	Historically, UPS has not developed logistical IT solutions internally	**Likely choice**	**May be opportunity to acquire part of the end-to-end solution**

IT Summary

- No integrated, scaleable IT solution exists today
  - functional pieces exist amongst “best of breed” vendors
  - multiple vendors sense the market opportunity and are pursuing an integrated solution
- Most realistic options to get to market
  - custom integration
  - purchasing integrated subcomponents
- First pass at cost and timing indicate IT is not a deal breaker
  - Cost: ~$30M
  - Timing: 12-18 months

Promising Horizon:

Length and Buckets:

The promising horizon is the length of time into the future that real-time promising is able to execute. The requirement is that a minimum of three months be available for promising. The buckets should be daily. In addition, as Demand Planning functionality is added in future business releases, the promising horizon should be extended by an additional three months for replenishment planning. These buckets could be weekly buckets.

Beyond the bucketed horizon, an infinite bucket needs to be created. This bucket will allow all orders requested outside the promising horizon to be promised assuming that inventory and capacity will be available. (See below for more details regarding processing of orders outside the promising horizon.)

A mechanism must exist to daily cycle the promising horizon. The "past" day will be deleted, and a "new" day will be added at the end of the horizon. For example, if the promising horizon is 21 days long, and today's date is June 12th, the last day of the horizon will be July 2nd. At some pre-determined point (probably midnight), the June 12th date will expire and fall out of the promising horizon, and the dates will range from June 13th through July 3rd.

Promising Outside the Promising Horizon:

If an order is received outside the daily promising horizon buckets, it is assumed that there is infinite capacity and inventory to promise the order. (If this assumption was not made, the order would be rejected and sent back to the customer without a promise. This is an unacceptable response.) The order is then queued in the "infinite" bucket that exists outside the promising horizon.

As the cycling process occurs, orders will move into thè promising horizon based on the requested date by the customer. The re-promising process (see section below called **Promising Functionality: Re-Promising Process and Notification**) will have to validate that there is in fact inventory and capacity available on the request date to satisfy the order. If there is not, there must be an alert to notify the appropriate function that the order cannot be fulfilled. The order can either be pushed back into the infinite queue or left on an error log until the problem is resolved.

Promising Functionality:

General Functionality:

Promising is composed of several steps. It is initiated by a Customer's desire to purchase product. If the shopping experience is web based, the Customer starts the process by generating a request for a specific product and checking for availability. The quote is the result of a Customer's request for specific product(s) with a specific delivery date or to be shipped via a specific method. The quoting function will provide a list of options that meet the criteria of the request. The Customer will review the quote. Once an option is selected, a promise is generated that will reserve the inventory and capacity for the order. The order must then be persisted within the application.

Order Receipt Mechanism:

It will important for the application to accept orders for promising from different sources. The first source will be via a Client's web. These orders are managed in real-time, and include both a quote and a promise.

An additional mechanism for accepting orders is via flat file or EDI in a batch mode. These orders will generate a promise only.

Promising Rules if no Inventory Available:

If a Customer request is made and there is no available inventory to satisfy the rèquest, then the order promising functionality must be able to promise based again standard lead times or rules. Ideally, these would be established on a client by client basis since each client may have a standard lead-time with their suppliers that will vary by client. If this is not possible, the best alternative would be to push the promise out to the infinite bucket. The downside of this solution is that it may significantly extend the promise date.

Regardless of which method is used to promise without inventory availability, an alert needs to be generated to indicate that inventory is not available, so that action can be taken to notify the client that there is insufficient inventory in the system.

Promising Rules if no Capacity Available:

If a Customer request is made and there is no capacity available to satisfy the request, then the order would be pushed to the infinite bucket. An alert needs to be generated to indicate that sufficient capacity is not available within the promising horizon, so that action can be taken to add incremental capacity or source orders from other warehouse locations.

Re-Promising Process and Notification:

If the supply and/or demand picture changes, then there will be a need to recalculate the pegging of orders against inventory. This is especially critical if there is a supply decommit or an inventory discrepancy in the warehouse. These events could potentially delay the shipment of an order. A third potential source of re-promising of orders is a change in warehouse capacity. If the warehouse is unable to meet the order commitments on a particular day, the balance of unshipped orders will have to consume capacity on a subsequent day. This could potentially have a domino effect, pushing out orders due to lack of warehouse capacity.

The frequency of the recalculation will depend on how frequently the supply and demand vectors change. It is also dependent on the frequency with which warehouse capacity fluctuates. The more frequent the changes to these variables, the more frequently the recalculation should be done. The frequency will also depend on the length of time that it takes to do the recalculation. The longer the recalculation takes, the fewer times it should be executed.

A critical piece of the recalculation is notification of any late shipments/deliveries. This information must be communicated internally to e-Logistics. Additionally, the information must be communicated to the Client. Ideally, this would be sent via flat file (or some other agreed upon format) to the Information Delivery queue for the Client.

Quote/Promise Maintenance:

Partial Shipment:

In some instances, an order may not ship complete. This should happen rarely, because the existing Oracle OMS application will not pick release orders to EXE (the warehouse management application) unless sufficient inventory exists within the warehouse to ship the order complete. However, if an inventory discrepancy is discovered in the warehouse at the time the order is being picked and prepared for shipment, the promising application needs to be able to handle the partial shipment.

EXE will have to update the to reflect the order quantity to be the same as the shipped quantity. This information will be provided to the promising engine as a ship confirmation (either directly from WMS or indirectly from another application). The promising engine needs to be able to recognize that there is a balance on the order that has not shipped. The process of providing the ship confirmation should close out the completed part of the order, and release the balance for re-promising.

Promise Change:

Once an order has been promised, the order is persisted in the Oracle OMS system as well as the promise engine. Prior to shipment the order will be "frozen" to prevent the order from being changed while it is being executed in EXE. The timing of the "freezing" of the order prior to shipment will be a business decision made by e-Logistics based on release of orders to the warehouse.

During the time that an order has been promised and the time it is frozen for execution, it is possible for the customer to request an order change. This change needs to be captured via the promising engine and replicated in OMS, so that the order stays synchronized in both applications. The mechanism for changing the promise and subsequently the order, needs to be determined.

Promise Cancel:

During the time that an order has been promised and the time it is frozen for execution, it is possible for the customer to request an order cancellation. Once the order is frozen, it will go through the execution process and ship to the Customer. (This business rule assumes that in this situation, the order is likely to generate a return. However, this is preferable from an execution standpoint. Trying to track down the order once it has been released to the warehouse creates a significant amount of extra work.)

If a cancellation is received during the appropriate timeframe, it needs to be captured via the promising engine and replicated in OMS, so that the order stays synchronized in both applications. The mechanism for cancelling the promise and subsequently the order, needs to be determined.

Table Of Contents

Document Control i
Change Log i
Related Documents i
Author Contact Information i

Table Of Contents ii

Overview 1
Overall Workflow 1
Technical Issues 2
Guiding Principles 2

Inventory Available To Promise (ATP) 3
Supply Components 3
Demand Components 3
Allocation Process 4

Warehouse Capacity ATP 5
Allocation Process 5
Consumption 5
Maintenance/Adjustments 5
Daily Bucket Definition 5

Promising Functionality 6
General Functionality 6
Order Receipt Mechanism 6
Quality of Promise 7
Promising Rules if no Inventory Available 7
Promising Rules if no Capacity Available 7
Re-Promising Process and Notification 8
Future Requirements 8

Promising Horizon 9
Length and Buckets 9
Promising Outside Horizon 9

Value Added Services (VAS) 10
Definition 10
Capacity Allocation 10
Inventory Items 10

Promising Rules 11
"Bounded" Promises 11
"Unbounded" Promises 11
Shipment Options 11
Promising When Not Managing Outbound Transportation 11

Delivery Network 12

Promise Maintenance 13
Partial Shipment 13
Promise Change 13
Promise Cancellation 13

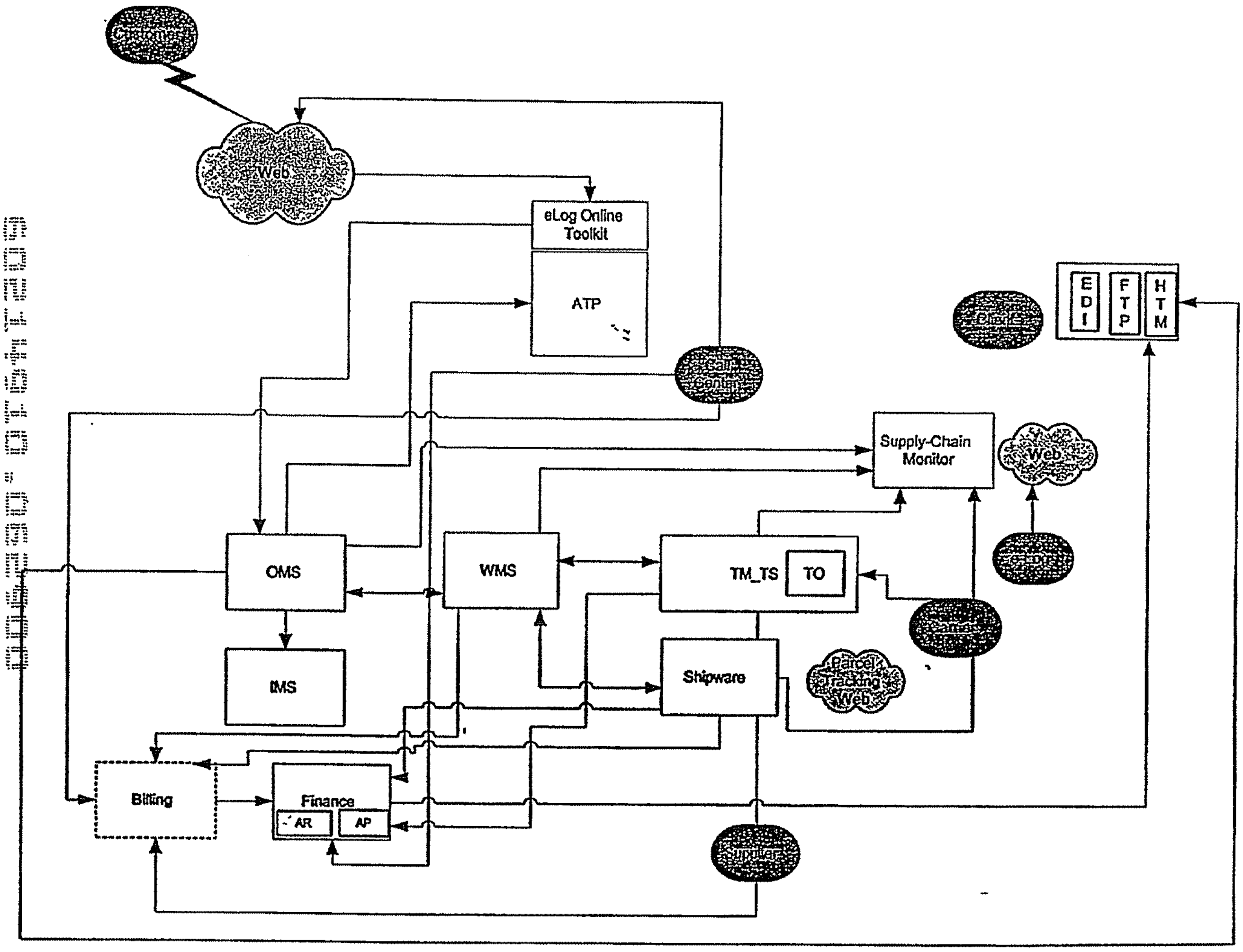
Appendix A: Glossary 15

Overview

This document describes the business requirements for the order promising features of the e-Logistics (e-Log) supply chain management solution. These are described in the chapters that follow.

Overall Workflow

It is necessary to understand the general workflow associated with the overall solution. This is shown in the diagram below. It is beyond the scope of this document to describe this diagram in detail; it is included here for reference purposes. Note that this diagram is a generalized workflow that does not reflect the design work that will occur as the promising solution is developed.



Technical Issues

Understanding how the e-Log client interacts with the e-Log solution is critical to understanding the full set of promising and quoting requirements.

A client's back office system, including the client's customer-facing Web presence, interacts with e-Log in two ways. For simplicity, e-Log provides the ability for batches of orders to be received via legacy flat file methods such as FTP and EDI. As these batches arrive, the orders are entered into the e-Log Order Management System (OMS) and responses are sent back to the client as necessary.

In addition, a client's Web site may interact with e-Log in real time via a set of Web application interfaces. The three interfaces e-Log is building are eLog.Order, eLog.Promise, and eLog.Quote. Ideally, a client would execute all its orders with these methods, enabling e-Log to capture orders at the earliest possible time and thus commit the tentative reservations made with eLog.Promise very quickly.

Even though orders may be received through both of these external interfaces, e-Log's back office is expected to have only one entry point for orders. From a back office point of view, therefore, all orders arrive one by one and are therefore processed in real time.

Guiding Principles

There are some overriding principles that guide e-Log to these business requirements.

All Orders Promised – Although some e-Log clients may not use the real-time promising features of the system, every order taken by e-Log must actually be promised. This is the only way that the promising engine can obtain full supply chain visibility and make promises correctly. Note that it is possible to take an order via the Web interface eLog.Order *without* having first used eLog.Promise because a promise is made or confirmed whenever an order is posted.

Uncommitted Promises Expire – Making a promise via the real-time eLog.Promise does not mean that e-Log has captured the associated order. Therefore, even orders that have a promise attached when they arrive at e-Log's OMS must be communicated to the promising engine again so that a firm reservation can be made. Unless this firm reservation is obtained, pending promises will expire within some reasonable time frame (current thinking is 5 to 30 minutes). By the same token, orders arriving with a promise but after that promise has expired can simply obtain a new promise and communicate that back to the client.

WMS Limitations – The chosen warehouse management system (WMS), Exceed 4000, has certain limitations that force postponement of some desired features. Because these limitations affect order management, they also create certain promising problems and challenges. This document calls those out and suggests ways to handle them.

Fastest Possible Processing – A goal of the system is to process all transactions as rapidly as possible. Unless prevented by application limitations, transactions should not be delayed. In particular, each contributing application should receive whatever information is relevant to it as early as it can be delivered. Real-time processing of single transactions is preferred to any type of batch operation.

Promising Mechanism – The e-Logistics business rule for the promising process is that a Client either places orders via the Web interface or via batch flat files delivered via FTP or EDI, but not both. Therefore, real-time promising is not available for clients delivering their orders via one of the batch modes. In other words, if the client Web site wishes to use the eLog.Promise Web interface, it must deliver that promised order via the eLog.Order Web interface. Orders received via the batch modes are promised internally at the same time the order is posted.

Inventory Available To Promise (ATP)

Supply Components

A Client may source the same product in multiple warehouses. Based on the client defined business rules, the ATP inventory associated with each warehouse may be a potential source of supply to fulfill an order, based on the rules specified in the Shipping and Delivery Rules sections. Multiple clients may source the same product stored at the same warehouse. Each client's inventory must be maintained independently.

Inventory will exist in various statuses, or states of availability. Some of those statuses will not be considered in ATP. The promising application must be able to differentiate which statuses are "netable" and which are not.

ATP will consider inventory receipts from supply orders once they are released into an available status. The inventory receipt must decrement the associated supply order or ASN so that the outstanding supply vector is accurately reflected.

Adjustments to inventory may occur as a result of cycle counts that indicate inventory inaccuracies and reconciliation of deliveries. These adjustments will be reflected in the inventory updates. The adjusted inventory will be made available/unavailable for allocations and promising as fast as possible.

In some application release after BR1, the inventory picture must begin to include information beyond the "4 walls" of the warehouse. Supply orders and Advanced Ship Notifications (ASN) will be incorporated as part of the inventory picture, in the appropriate daily bucket based on anticipated receipt date. If both of these components (supply orders and ASN's) are to be included, then the ASN must be decremented from the appropriate supply order so that anticipated supply is not overstated.

Collaborative supply planning is a requirement in a future business release. This functionality needs to provide an extended view into the supply chain, by looking at either the supplier's finished goods inventory levels or capacity to produce inventory. This view needs to be available to the Client and their supply base.

Returns occur for a variety of reasons. Depending on the reason and the condition of the goods being received, the product then becomes a part of the inventory for consideration in the promising process. This inventory adjustment will be made available for promising.

Demand Components

A request for a promise may consist of single or multiple order lines. The primary data elements input by the Client are item, item quantity, customer request date, delivery options, shipping option, customer zip code or postal code, and a client-unique ID code. The promising engine holds inventory for a period of time upon the expectation of an order; the hold is removed after the time is expired. The promising engine must be able to identify a re-promise for an existing promise so that duplicate holds are not placed, the purpose of the client-unique ID code.

An order may consist of single or multiple order lines. The primary data elements input by the Client are item, item quantity, customer request date, delivery options, shipping option, customer zip code or postal code, and a client-unique ID code. The order engine posts the order and obtains a promise that reserves inventory or upgrades a hold to a reservation if a promise has previously been issued (using the client-unique ID code).

The order and promising functions must provide for both international and domestic orders, although international orders are not part of BR1.

Quotes contain the same data elements as promises. However, a quote does not reserve or hold inventory. The purpose of quotes is to provide a range of options for the landed cost of the order (e.g., 2-day air vs 4-day ground). Based on e-Logistics' current understanding of available promising technologies, it is anticipated that promising can only return a single, optimized option. This makes quoting less interesting for the Client and probably means that quoting will not be available until a later business release. It is important to note that a promise taken after a quote may yield a different result because the quote does not hold inventory and time will have elapsed between quote and promise.

Collaborative demand planning is a requirement in a future business release. Collaboration will provide the ability for a Client and their customer base to review forecasts and edit them. Additionally, the view needs to be available to e-Log for providing Demand Planning services to their Client base and for optimization of internal operations.

Allocation Process

The Client's inventory will be maintained in daily buckets and allocated on a first come first serve basis.

The Client's inventory will not be segmented by customer or customer group in the initial release. In future releases, segmentation will be required and inventory allocation will include the segmentation with all other allocation rules.

Warehouse Capacity ATP

Allocation Process

A mechanism is needed to represent warehouse capacity for each warehouse. Warehouse capacity must reflect the warehouse resources available at the time for normal fulfillment of orders as well as the separate warehouse resources used to provide value added services (see the Section on VAS). Warehouse capacity must be represented in daily buckets and span the length of the promising horizon.

e-Logistics needs the flexibility to allocate all warehouse capacity on a client by client basis, allocate a portion of the capacity to clients with the balance in a general pool, or have no allocation of warehouse capacity. The promising engine must be capable of handling all these types of capacity allocations. Client by client allocation is especially important, and may prevent capacity pooling in early business releases, because e-Logistics may be contractually obligated to allocate specific warehouse capacity to a specific client. Therefore, the capacity model must be flexible, easily maintainable, and meaningful to the promising engine.

If specific clients have been allocated specific warehouse capacity, there must be a way to release excess capacity for general consumption by other clients. A failure to release unused capacity can quickly create inefficiencies in the warehouse and thus must be avoided.

The promising engine must support a set of special order types in which warehouse capacity constraints are either not considered or are applied differently from the standard promising rules. The two known types (there could be others) are "rush" and "economy." A rush order, a premium service, will ship immediately as long as the order is received by the cut-off time and inventory is available. An economy order will ship without a guaranteed arrival date; the warehouse must ship the order no later than a given date but may ship at its convenience before that date.

Consumption

Unallocated warehouse capacity for a given pick date will be consumed on a first come, first serve basis. A specific client's allocated warehouse capacity for a given pick date will be consumed only by that client until the e-Logistics business rules allow the client's unused capacity to be de-allocated and added to the pool. A client with allocated warehouse capacity must consume all of its allocated capacity for that particular day before it is allowed to consume capacity from the unallocated pool.

Maintenance/Adjustments

Because the physical capabilities of a warehouse may change (e.g., a warehouse becomes automated, specialized goods require more handling, lift truck breaks down, a client expands), it must be possible to adjust the capacity constraints so that the promising engine is accurate. The system must be capable of maintaining separate allocations for such things as specific clients, categories of clients, categories of goods, and general warehouse resources.

Daily Bucket Definition

A mechanism must exist to cycle the warehouse capacity ATP daily. The "past" day will be deleted, and a "new" day will be added. (See example in Promising Horizon.) Unlike inventory buckets that can use the standard 24-hour clock for ATP, the warehouse capacity must use a slightly different clock. It needs to be driven by the cut-off times for meeting carrier pick-ups. For example, if the last pick up of the day is 7pm, then at that time the capacity ATP rolls over, reflecting the following day's date as the earliest possible date available for order promising. This roll over must be warehouse specific and take into account the appropriate time zone and cut-off considerations.

Promising Functionality

General Functionality

A request for an e-Logistics promise is initiated by an e-Logistics client's Web site via the e-Logistics Web interfaces. The request is based on the client's interaction with its customer. The nature of the request, such as asking for a specific delivery date or simply asking for options, is based on the client's Web site and e-commerce capabilities. Once the client has collected the appropriate information, the promise request is transmitted to e-Logistics.

The promising engine constructs a promise based on e-Logistics business rules, client configuration, inventory availability and warehouse constraints as defined by this document. In addition, the promising engine places a hold (a temporary reservation with an attached expiration timer) on the inventory associated with the promise. The response is then transmitted back to the client. Note that the returned promise may not precisely match the request. For example, the request may have specified a specific delivery date but the best promise e-Logistics could make is later than that date.

The client takes the response to the promise request, alters it if necessary (e.g., marks up shipping costs), and presents the result to the customer. If the customer approves and commits to an order, the client will transmit that order via the Web interface.

There must be a client-unique identifier assigned to the promise. This identifier must be used with the initial promise, any re-promises of the same promise, the order associated with the promise, and any subsequent order changes. This is necessary to prevent multiple holds and/or reservations of the same item(s).

If a Customer orders multiple line items with different delivery addresses, these are considered to be separate orders and will be promised as such. The Client must be able to split these into unique orders by delivery address before sending to e-Logistics for promising. Note that this is a client activity; the client's customer may be unaware that this is happening.

A promise will not always meet the original date requested by the Customer. The original request date will be captured during the promising and/or order receipt process, and monitored as order re-promising is executed. The functionality is required to support optimizing an order by trying to meet the original requested date. If a change in the supply/demand picture frees up inventory, the e-Logistics system must be able to improve the promise by moving it as close to the original requested date as possible.

Although quoting is not part of the initial Business Release of Order Promising, it will be necessary in future business releases. The development of the promising engine must not preclude adding this functionality at a later time. In particular, when Configure to Order (CTO) is added, quoting will become a critical component of Order Promising.

Order Receipt Mechanism

It will be important for the application to accept orders for promising from different sources. The first source will be via a Client's web. These orders are managed in real-time and may be preceded by quotes (future release) and promises. Orders may also be accepted via flat file or EDI in a batch mode. These orders will generate a promise as they are posted.

The quote process is not mandatory. Particularly in a B2B environment, the Client's customer may not be interested in getting quotes but only in obtaining the promise information. The Order Management application needs to be able to accept an order for which promises have not been obtained.

As each order is received by OMS, the order is matched against the promising engine. If a promise exists, that information will be matched up against the order and the temporary hold of inventory and capacity will be firmed up as a reservation. If no promise exists, the promise engine needs to verify that inventory and capacity are available, and respond to OMS with a delivery date. OMS must capture this information and return it to the Client.

Quality of Promise

Initially, promising will be based on the inventory within the "four walls" of the warehouse. In subsequent releases, visibility of the supply chain will extend the order promising functionality to include PO's, ASN's, and potentially finished goods inventory at other locations including the Client's suppliers.

Because these extended elements of the supply chain probably will not be under e-Logistics' control, promises based on the broader supply chain may not be as dependable as those where e-Log does have control. Therefore, it is necessary to communicate information back to the client indicating in which part of the supply chain the promise falls. It falls to the client to determine how to communicate this information to its customers. For example, a client might decide that a promise based on a purchase order is not as strong as one based on available inventory or inbound goods and therefore add an extra day to the promise when it is sent to the customer.

The following levels of quality have been identified so far. The table shows how a client might interpret the quality:

Supply Chain Location	Promise Quality
"Four Walls" Inventory	Guaranteed
ASN	Very High
PO (Supply Order)	High
Other Supply Source	Medium

Promising Rules if no Inventory Available

If a request for promise is made and there is no available inventory to satisfy the request, items will be promised using the Client's rules; initially, this will probably mean using the standard lead times from the item master or returning a "no promise". If the Client wishes to promise based on standard lead times, the client must provide this information as part of the item master file. If the Client business rule is that a "no promise" is returned, then it is up to the Client to provide information regarding availability to its Customer. It is not possible for the e-Logistics system to accept an order under "no promise" circumstances, so the client's only option is to re-submit the order when the supply chain picture changes and enables a valid promise to be generated.

Regardless of which method is used to promise without inventory availability, an internal alert needs to be generated to indicate that inventory is not available, so that action can be taken to notify the client that there is insufficient inventory in the system.

Promising Rules if no Capacity Available

If a request for promise is made and there is no capacity available to satisfy the request, then the order would be pushed to the infinite bucket. An alert needs to be generated internally to indicate that sufficient capacity is not available within the promising horizon, so that action can be taken. Note that this is strictly an internal problem; the client will have been given a valid promise with specific dates and will not have any indication that the problem exists.

Re-Promising Process and Notification

If the supply and/or demand picture changes, then there will be a need to recalculate the pegging of orders against inventory. This is especially critical if there is a supply decommit or an inventory discrepancy in the warehouse. These events could potentially delay the shipment of an order. A third potential source of re-promising of orders is a change in warehouse capacity. If the warehouse is unable to meet the order commitments on a particular day, the balance of unshipped orders will have to consume capacity on a subsequent day. This could potentially have a domino effect, pushing out orders due to lack of warehouse capacity.

Re-promising will not automatically upgrade the transportation service level to try to maintain the original promise date. The transportation service level associated with the order at promising will remain valid, regardless of the cause of the delay. If a determination is made to upgrade the service level by e-Logistics, manual intervention will take place to make the change.

The frequency of the updating ATP and the promises affected by the change needs to be as real-time as possible. The ideal situation would be to update ATP with each transaction; e-Logistics expects every effort to be made to enable real-time promising. Currency of data is critical to an accurate real-time ATP.

When the re-promising is executed, the application needs to be capable of "honoring" those orders that have been frozen by the order management system (see the Promise Maintenance chapter). Frozen orders must not be considered in re-promising calculations because they are being fulfilled and cannot be changed.

A critical piece of the recalculation is notification of any late shipments/deliveries. This information must be communicated internally to e-Logistics. Additionally, the information must be communicated to the Client. Ideally, this would be sent via flat file (or some other agreed upon format) to the Information Delivery queue for the Client. As mentioned earlier, if the re-promising process is able to improve on an existing promise, and move it closer to the original request date, the change also needs to be communicated to the Client via information delivery.

Future Requirements

There are several future business requirements that need to be considered in design of the promising application so that e-Logistics is not precluded from adding them later.

- *Merge in transit*: coordination of multiple shipments to arrive at a single destination as a single shipment. This requires more that just the ability to ship multiple orders to arrive on the same date. The concept is that there is a merge point (hub) somewhere in the delivery network that allows consolidation of multiple shipments, potentially from multiple carriers.
- *Cross-Dock*: ability to recognize opportunities to move inventory directly from an inbound shipment to an outbound shipment without processing a putaway and a pick on the inventory. This will require
- *Flow Through*: merging of product within an e-Logistics warehouse for shipment to the end customer. This will require the ability to determine availability within the warehouse and determine additional availability within the supply chain to complete the order. Inventory will be transferred into the initiating warehouse and shipped as a single shipment to the Customer.
- *Drop Shipment*: ability to execute a shipment of inventory from a supplier to a Customer that never "lands" at an e-Logistics warehouse. This will require extended visibility into the supply chain, and connectivity to supply sources to execute the order.

Promising Horizon

Length and Buckets

The promising horizon is the length of time into the future that real-time promising is able to execute. The requirement is that a minimum of three months be available for promising. The buckets must be daily. In addition, as Demand Planning functionality is added in future business releases, the promising horizon must be extended by an additional three months for replenishment planning. These buckets could be weekly buckets.

Beyond the daily buckets, an infinite bucket must be created. This bucket will allow all orders requested outside the promising horizon to be promised assuming that inventory and capacity will be available. (See below for more details regarding processing of orders outside the promising horizon.)

A mechanism must exist to cycle the promising horizon daily. The "past" day will be deleted, and a "new" day will be added at the end of the horizon. For example, if the promising horizon is 21 days long, and today's date is June 12th, the last day of the horizon will be July 2nd. At some pre-determined point (probably midnight), the June 12th date will expire and fall out of the promising horizon, and the dates will range from June 13th through July 3rd.

Note: As noted above, the promising horizon cut-off for inventory ATP is midnight local time. The cut-off for capacity will be set as a single time by warehouse, and will coincide with the cut-off by carriers for pick ups. It will be critical to understand synchronization issues of promising during the window between the capacity date roll over and the inventory date roll over.

Promising Outside Horizon

If an order is received outside the daily promising horizon buckets, it is assumed that there is infinite capacity and inventory to promise the order. (If this assumption was not made, the order would be rejected and sent back to the customer without a promise. As a matter of e-Logistics policy, this is an unacceptable response.) The order is then queued in the "infinite" bucket that exists outside the promising horizon. The order is in "suspense" while in the infinite bucket.

As the cycling process occurs, orders will move into the promising horizon based on the requested date by the customer. This is the only priority associated with the order. Once it moves within the promising horizon, the re-promising process and inventory will have to validate that there is in fact inventory and capacity available. A promise is then generated and the information is passed to the Client. In this situation, no promise is created with the initial request. Inventory and capacity are not reserved. The reservation occurs when the promise can be executed. The Client then needs to be notified, so the order can be sent to e-Logistics. If there is not sufficient inventory or capacity, then there must be an alert to notify the appropriate function that the order cannot be fulfilled. The order can either be pushed back into the infinite queue or left on an error log until the problem is resolved.

Value Added Services (VAS)

Definition

VAS is performed on both inbound and outbound shipments by e-Logistics for the Client. This is not an exhaustive list, but a representative example of some of the potential requests from Clients. They can include a variety of services:

- Inbound product inspection
- Repackage of product
- Insertion of catalogues
- Kitting and assembly
- Gift wrap

These services can be provided on a Client level, order level, or line item level.

Capacity Allocation

A separate work center needs to be established for each warehouse to manage capacity constraints for VAS. When VAS are promised, the warehouse capacity at these work centers needs to be consumed, not the general warehouse capacity that is available for the fulfillment of orders.

In subsequent business releases, there needs to be the capability of adding multiple work centers for different types of VAS. VAS items will be tied to a specific work center through a routing, or other mechanism. In addition, the application needs to be capable of handling multiple work centers for a single item on an order. For example, an item may need to be assembled in one step and gift-wrapped in a second. There need to be mechanisms available to accomplish this (BOM's, routings, etc.).

Inventory Items

VAS items may have unique item numbers that will appear on the order. These could potentially include services like gift-wrap, catalogue insertion, etc. The items need to appear on the order so that the order is fulfilled correctly.

The lack of inventory on a VAS item may or may not affect fulfillment of the order. By default, lack of inventory of a VAS item will delay the order and push out the ship date. However, there are some situations in which the lack of inventory of the VAS item is not critical to the shipment (e.g., the warehouse is out of stock for a promotional flyer). For any VAS line item that is not critical, the promising engine must not consider inventory when deriving the promise.

Promising Rules

"Bounded" Promises

A bounded promise is a promise where the Customer will establish a parameter that the promising engine must consider in creating a promise. There are two parameters that might be provided by the Customer via the Client's web site: either a Date Specific Delivery (DSD) request or a specific method of shipment/service level. In either case, the promising engine needs to "solve" the promise by determining a second variable. When the DSD is provided, the promising engine must determine the transportation service level required to meet the requested date. When the transportation service level is provided, the promising engine must determine what the promise date will be.

"Unbounded" Promises

When the Customer does not specify a DSD or a specific service level, the promise is considered "unbounded." The promising engine will return a promise date based on standard transportation service levels (typically ground for parcel, and standard lead-time for LTL and TL).

Shipment Options

The Client/Customer has two options for each shipment. All items on the order can ship from a single warehouse on the same day via the same transportation mode, or line items can be sent on multiple dates, multiple warehouses, or a combination of both.

When a single shipment policy is used, the shipment will occur on the availability date of "latest" available item. The single shipment tends to be the least expensive from a transportation standpoint.

Multiple shipments must always try to ship as much of the order as possible from the closest location (assumed to be the least expensive option). The balance of the order must be sourced from the next closest facility, etc.

The promising engine must satisfy whatever parameters requested by the Customer first, and then solve the request in the least expensive method possible. The delivery network must provide information to accomplish this. (See the section on Delivery Network for further information.)

Promise Type	Information Specified	Solve For
Bounded	Date Specific Delivery (DSD)	Transportation Mode
Bounded	Transportation Service Level	Promise Date
Unbounded	N/A	Use "Standard" Transportation

Promising When Not Managing Outbound Transportation

If e-Logistics is not managing the outbound transportation for a Client, then the promising capability will be limited. The promising engine will return only a ship date to the web site. Because the delivery network will not be established for that Client, it will not be possible for the promising engine to determine transit time and manage pick up and delivery of the order. (See the Delivery Network section for more information.)

Delivery Network

The promising software needs to be able to handle a complex delivery network with multiple lanes, multiple zones, and a pricing structure for the lanes. The zones need to be defined by zip codes and the delivery network has to provide sufficient modes of transportation to travel within those zip codes (also see delivery service level below). For example, there could be UPS ground, FedEx Overnight, mail Five-day, etc. It also needs to be able to handle domestic zip codes as well as international postal codes.

The Delivery Network also needs to be maintainable. It must be easy to add, remove, delete, or change cost structure, delivery lanes, shipping zones, and modes of transportation. Ultimately, e-Logistics needs to be able to maintain unique networks for each Client.

The delivery network will be implemented in phases. Below is the outline of the requirements by phase. These are available only to Clients utilizing e-Logistics Outbound Transportation services.

Phase	Description	Features
I	Promising	• Ship Date Only • Delivery Date (UPS only)
II	Promising w/Cost and Delivery Date	• UPS • TS (For both core carriers and Client specific contract rates) • International (Transportation cost only)
III	International	• UPS Total Landed Cost included customs and duties • TS Total Landed Cost included customs and duties

In the initial release, the delivery network will be relatively simple. Ship dates will be returned on all requests based on inventory and capacity availability. It will be the Client's responsibility to translate this information into a delivery date. The only exception to this would be on orders using UPS as the carrier. The delivery network with transit times either needs to be imbedded into the Global ATP delivery network, or it needs to be accessed real-time to provide the delivery date. The return message to the Client will always include the ship date and a N/A for the delivery date with the exception of UPS shipments.

In Phase II, the delivery network becomes Client specific and provides both cost and delivery date information on all Client's promises where the outbound transportation is managed by e-Logistics. e-Logistics needs to be able to create each network based on the Client's rates with carriers, using either the TS core carriers or their own specific contract rates. These networks need to be either incorporated into Global ATP or accessed via internet sites (or other mechanism) so that the information can be provided during order promising. There may be multiple sources for this information. The additional communications overhead associated with accessing external information sources may introduce additional latency that might affect the real-time performance of the e-Logistics solution; this must be considered when designing this solution.

Phase III of the solution includes more sophisticated management of International orders, and must include total landed cost for international shipments including transportation, customs, and duties.

For order promising requests that are not interactive, or for cost information that cannot be accommodated in real-time, mechanisms need to be available to pass the cost information back to the Client (via Information Delivery), to the Customer (via e-mail), or to both.

Promise Maintenance

There are limitations in the current functionality of EXE's EXCeed 4000 system, the warehouse management tool that is driving some of the solution described below. First, EXE is not capable of showing a short shipment or a backorder. This deficit requires that in a situation where an order is shipped short, that the EXE order be adjusted to show that the shipped quantities and ordered quantities match. This impacts the process below described as "Partial Shipment".

Secondly, the OMS and WMS systems are not designed to maintain synchronization of orders. The system has been designed for OMS to maintain the orders and pass them to WMS for execution. This architecture restricts the view of WMS to the queue of orders for planning purposes.

Finally, one of the e-Log business requirements has been that order maintenance was the responsibility of the Client, who would maintain the order and release it at the time of shipment. Because the new architecture can handle many order management tasks, e-Log now wants the client to deliver the order immediately. This means that the responsibility for maintaining the order now lies with e-Log. The new requirement for e-Log to maintain orders and the lack of synchronization between OMS and WMS are the two factors that contribute to the process described below regarding freezing of orders just prior to release to WMS.

differen. ←

Partial Shipment

In some instances, an order may not ship complete. This should happen rarely because the existing Oracle OMS application will not pick release orders to EXE (the warehouse management application) unless sufficient inventory exists within the warehouse to ship the order complete. However, if an inventory discrepancy is discovered in the warehouse at the time the order is being picked and prepared for shipment, the promising application needs to be able to handle the partial shipment.

EXE will have to update the order so that the order quantity is the same as the shipped quantity. This information will be provided to the promising engine as a ship confirmation (either directly from WMS or indirectly from another application). The promising engine needs to be able to recognize that there is a balance on the order that has not shipped. The process of providing the ship confirmation closes out the completed part of the order and releases the balance for re-promising.

Promise Change

Once an order has been promised, the order is persisted in the Oracle OMS system as well as the promise engine. Prior to shipment the order will be "frozen" to prevent the order from being changed while it is being executed in EXE. The timing of the freezing of the order prior to shipment will be a business decision made by e-Logistics based on release of orders to the warehouse.

During the time that an order has been promised and the time it is frozen for execution, it is possible for the customer to request an order change. This change needs to be captured via the promising engine and replicated in OMS, so that the order stays synchronized in both applications. The mechanism for changing the promise and subsequently the order, needs to be determined.

Promise Cancellation

During the time that an order has been promised and the time it is frozen for execution, it is possible for the customer to request an order cancellation. Once the order is frozen, it will go through the execution process and ship to the Customer. (This business rule assumes that in this situation, the order is likely to generate a return. However, this is preferable from an execution standpoint. Trying to track down the order once it has been released to the warehouse creates a significant amount of extra work.)

If a cancellation is received during the appropriate timeframe, it needs to be captured via the promising engine and replicated in OMS, so that the order stays synchronized in both applications. The mechanism for canceling the promise and subsequently the order, needs to be determined.

Appendix A: Glossary

Client: The e-Logistics customer who contracts with e-Logistics for services.

Customer: e-Logistics Client's customer